Problem Set 1

Michelle Newcomer

Stat 242 Fall 2013

Problem 1

Class survey: done

Problem 2

a) I cloned the Git repository onto my machine using Github for Windows.

b) Commits made to my own repository on Git.

D:\Users\Michelle Newcomer\Documents\GitHub\LecturePractice [master]> git log commit c02baebc3597c4eeec4b80e78221bbdccde760fc Author: Michelle Newcomer mnewco8290@yahoo.com Date: Thu Sep 12 18:26:00 2013 -0700

```
Practice with Unit 3
```

commit 3380b7ad17fecf409aa93affb799d744346a6aa0 Author: Michelle Newcomer mnewco8290@yahoo.com Date: Mon Sep 9 09:15:58 2013 -0700

```
Practice with Unit 1 and Unit 2
```

commit 47eb8e3ee64583345b8ce38f319e89c84510689e Author: MNewcomer mnewco8290@yahoo.com Date: Fri Sep 6 10:22:44 2013 -0700

```
Lecture practice created
```

D:\Users\Michelle Newcomer\Documents\GitHub\LecturePractice [master]>

Problem 3

Throughout the decades the top producer of apricots has shifted from Lebanon in 1965 to Ukraine in 2005.

```
#! /bin/bash
echo "Please enter the code # of the type of crop then press enter:
"
read input_variable
echo "You entered: $input_variable"
IFS=: # internal field separator
type=$input_variable # this specifies the type of crop
```

```
# the wget grabs the zip file and the -O allows the name to be
changed
wget -O Data${type}.zip
"http://data.un.org/Handlers/DownloadHandler.ashx?
DataFilter=itemCode:"${type}"&DataMartId=\
FAO&Format=csv&s=countryName:asc,elementCode:asc,year:desc&c=2,3,4,5,6,7&"
 #this line downloads the data and renames the data to Apricots.zip

# this for loop extracts each zip file and appends .csv to the
original zip file name
for i in *.zip #this line counts the number of zip files
do
  n=$(unzip -lqq $i | awk '{print $NF}') #-l unzips the files names
and -qq does it quietly while only giving the file names inside the
zip to the screen. This then pipes the information to awk with
searches the output to find the name of the fields
    e=${n#*.} # this line looks at the file name inside of the zip
file and removes everything in front of the . and saves the csv
portion
    unzip $i && mv $n ${i%%.*}".$e" #this line unzips the file and
renames the unzipped file to be the original portion plux csv
    rm $i #This then removes the original zip file because it has
already been extracted
done

sed 's/, / /g' *.csv > UN_No_Comma.csv #this removes the comma
sed 's/\"//g' UN_No_Comma.csv > UN_No_Quote.csv #this removes the
extra " in the country name
grep -i + UN_No_Quote.csv > UN_World_Regions.csv #this separates the
regions from the countries and save the regions
grep -i -v + UN_No_Quote.csv > UN_Countries.csv #this separates the
countries and saves countries

yrs=1965:1975:1985:1995:2005

for yr in $yrs
do
    grep -i -e ${yr} UN_Countries.csv > UN_Countries_${yr}a.csv
#this grabs entries that have the year pattern
    grep -i -v ${yr}.[0-9] UN_Countries_${yr}a.csv >
UN_Countries_${yr}b.csv #grabs the countries that do no have the
pattern year. because that means it is a double and represents the
acreage
    rm UN_Countries_${yr}a.csv
    grep -i -e Area -e Harvested UN_Countries_${yr}b.csv >
UN_Countries_${yr}c.csv #grabs only the entries with words area and
harvested
    rm UN_Countries_${yr}b.csv
    sort -t',' -k6 -r UN_Countries_${yr}c.csv | head -5 >
${yr}_top5.txt #sorts the 6th column, puts it in reverse order, then
pipes the output to a text file
    rm UN_Countries_${yr}c.csv
    cat ${yr}_top5.txt
done
```

Problem 4

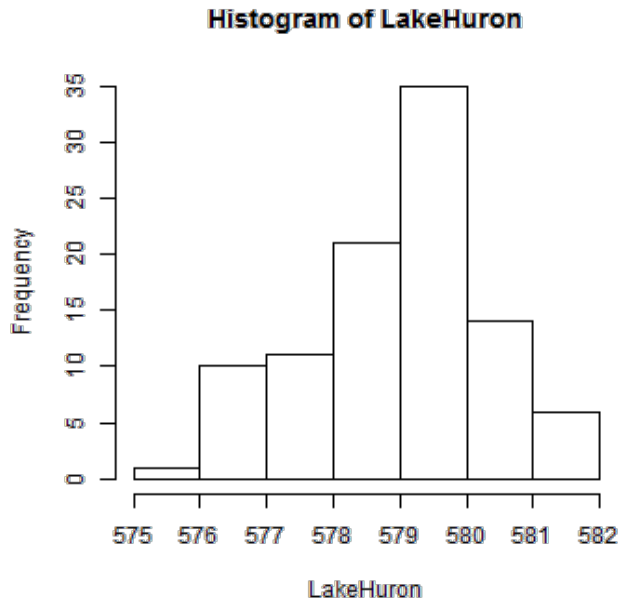
```
#! /bin/bash
wget -O index.txt http://www1.ncdc.noaa.gov/pub/data/ghcn/daily/
grep -no 'a href="[^"]*.txt"' index.txt > names.txt #this looks for
the pattern a href = textfile name
sed 's/\"//g' names.txt > names_no_quote.txt #this removes the extra
" in the name
awk -F '=' '{print $2}' names_no_quote.txt > names_only.txt #save
only the text after the = sign

i=$(wc -l names_only.txt) #this gathers the number of lines
i=${i% *} #this gets just the number from the output of wc
lines=$(seq 1 ${i}) #this creates a vector from 1 to i

for num in $lines; #this for loop looks at the first line and
download the first text file and tells the use wich file is
downloading then iterates.
do
    c=$(sed -n ${num}p names_only.txt)
    wget -q ${c} http://www1.ncdc.noaa.gov/pub/data/ghcn/daily/${c}
    echo "Downloading the" ${c}" file"
done
```

Problem 5

```
hist(LakeHuron)
```




```
lowHi <- c(which.min(LakeHuron), which.max(LakeHuron))
yearExtrema <- attributes(LakeHuron)$tsp[1] - 1 + lowHi
```

The lake levels have fluctuated over the years. The lowest level was recorded at 575.96 in 1964. The highest level was recorded at 581.86 in 1876.